

Xenv

version 4.1, 23 February 2023

Sergey Poznyakoff.

Copyright © 2021–2023 Sergey Poznyakoff

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover, and no Back-Cover texts. A copy of the license is included in the section entitled “GNU Free Documentation License.”

Table of Contents

1	Overview	1
2	Variable references	1
3	Escaping special characters	2
4	Command substitution	3
5	Comments and verbatim quotations	3
6	Meta-variable	3
7	Directives	4
7.1	Diagnostic messages and forced exit	6
7.2	Looping directives	7
7.3	Eval	8
7.4	Diversions	9
8	Invocation	10
8.1	Exit codes	13
9	Downloads and Installation	13
10	Bug Reports	13
Appendix A GNU Free Documentation License		
	13
	Index	22

1 Overview

`xenv` reads input files and prints on the standard output their contents, replacing references to the environment variables with their values and shell commands with the output they produce.

It can be used, in particular, in docker containers where using environment variables for configuration became a de-facto standard. This method is obvious when the configuration is consumed by some programming language (PHP, Python or the like). It is difficult, however, to expand environment variables in configuration files of the programs that provide no mechanism for this (such as `mysqld` or `bind`, for instance). This is where `xenv` comes into play.

By default, `xenv` looks in the input text for the variable references and command invocations in POSIX shell format and expands them, reproducing the rest of the material verbatim. If the shell syntax does not suit well the language or structure of the input, an alternative syntax can be selected from the command line. A number of additional constructs is provided, such as comments, conditional directives, etc. These optional features can be controlled from the command line as well.

2 Variable references

A variable reference has the form ‘`$variable`’ or ‘`${variable}`’, where *variable* is the environment variable name. The two forms are entirely equivalent. The form with curly braces is normally used if the variable name is immediately followed by an alphanumeric symbol, which will otherwise be considered part of it. This form also allows for specifying the action to take if the variable is undefined or expands to an empty value:

`${variable:-word}`

Use Default Values. If *variable* is unset or null, the expansion of *word* is substituted. Otherwise, the value of *variable* is substituted.

`${variable:=word}`

Assign Default Values. If *variable* is unset or null, the expansion of *word* is assigned to *variable*. The value of *variable* is then substituted.

`${variable:?word}`

Display Error if Null or Unset. If *variable* is null or unset, the expansion of *word* (or a message to that effect if *word* is not present) is output to the current logging channel. Otherwise, the value of *variable* is substituted.

`${variable:+word}`

Use Alternate Value. If *variable* is null or unset, nothing is substituted, otherwise the expansion of *word* is substituted.

`${variable:|word1|word2}`

Display Alternative Values. Unless *variable* is null or unset, the expansion of *word1* is substituted, otherwise the expansion of *word2* is substituted.

In the constructs above the colon can be omitted, resulting in a test only for a variable that is unset.

In any expansion form, the *word* part is treated exactly as in Bourne shell: it can contain variable expansions, command substitutions, single and double-quoted parts. The latter two are interpreted as follows: the quotes are removed, the text between double quotes is subject to variable expansion and command substitution (see below), whereas the text between single quotes is not. Within double quoted part, a backslash can be used to escape the indirection character (`'$'`), double and single quote character and backslash itself.

Several command line options control the expansion process. Unless special reference forms are used, a reference to an undefined variable expands to an empty string. The `-r` option changes that. When given that option `xenv` will retain references to undefined variables in the produced output.

The `-u` option instructs `xenv` to treat a reference to an undefined variable as error.

If the input text can contain a considerable number of `'$'` characters, which can be misinterpreted as variable references, it is possible to change variable reference syntax to use a single *meta-variable* instead. For example, when invoked as

```
xenv -e ENV
```

`xenv` will recognize `ENV{name}` as the environment variable reference. The constructs `$name`, `#{name}` and others will be reproduced verbatim. This feature is described in detail in [Chapter 6 \[Meta-variable\], page 3](#).

3 Escaping special characters

A `'$'` preceded by a backslash loses its special meaning. For example, supposing that the variable `USER` is set to `'root'`, the string:

```
An escaped \USER is not expanded, unescaped $USER is.
```

will be expanded as follows:

```
An escaped \USER is not expanded, unescaped root is.
```

The `\` character must be escaped similarly:

```
Escape your \\ ⇒ Escape your \
```

This *escaping* can be turned off using the `-Wno-escape` option (see [\[feature control\], page 11](#)).

When expanding quoted strings in complex variable references, the following sequences are substituted with their second character: `'\$', '\\', '\", \'`. For example (assuming `VAR` is not defined):

```
#{VAR:-'quoted string'} ⇒ quoted string
#{VAR:-a lone \' character} ⇒ a lone ' character
```

These rules apply also if the default value is enclosed in double quotes:

```
#{VAR:-"Escape \\, \$ and \" in quoted context"}
```

The `-Wno-escape` option does not affect escapes in variable substitutions.

4 Command substitution

A construct

```
$(command)
```

is substituted with the output of the shell command *command*. `xenv` performs substitution by running `$SHELL -c command` and replacing the construct with the standard output produced by the command, with any trailing newlines removed. If the `SHELL` variable is not set, `/bin/sh` is assumed.

The *command* is passed to the shell verbatim. This means, in particular, that variable references in the command are expanded by the shell, rather than by `xenv` and, as a consequence, that the options `-u`, `-r`, and `-e` don't affect command substitutions.

By default, `xenv` will wait for as long as it takes for the command to terminate. It is, however, possible to limit command execution time using the `-t n` option. If this option is given, the command is permitted to run no longer than *n* seconds. If it doesn't terminate within that time, it will be forcibly terminated by sending it the `SIGKILL` signal, and a diagnostic message to that effect will be printed on standard error.

Command substitutions can be disabled using the `--wno-command` command line option (see [\[feature control\]](#), page 11).

5 Comments and verbatim quotations

Comments are multiline. They are introduced with the characters `#{*` and extend up to the nearest pair of characters `*}`. Comments are removed from the output.

Comments can be disabled using the `--wno-comment` option (see [\[feature control\]](#), page 11).

To reproduce a portion of text verbatim (i.e. without expanding variables and substituting commands in it), enclose it in `'$[` and `']` brackets. Newlines and balanced square brackets are allowed within that construct.

The special meaning of the `'$[...]`' construct can be disabled by the `--wno-quote` option (see [\[feature control\]](#), page 11).

6 Meta-variable

Environment meta-variable is a special feature, that replaces the traditional variable reference syntax (see [Chapter 2 \[Variable references\]](#), page 1) with referencing via a single meta-variable - an approach traditionally used in many programming languages. The name of the meta-variable is introduced using the `-e` command line option:

```
xenv -e ENV
```

Once the meta-variable is defined, all constructs discussed above lose their special meaning and are reproduced on the standard output verbatim. Instead, the following constructs are recognized as variable references:

```

$ENV{name}
$ENV{name:-word}
$ENV{name:=word}
$ENV{name:?word}
$ENV{name:+word}
$ENV{name:|word1|word2}

```

The constructs for command substitution, verbatim quotations and comments become `$ENV(...)`, `$ENV[...]`, and `$ENV{* ... *}`, correspondingly.

Any sequence of characters can be used as the meta-variable name, provided that the sequence itself is a valid variable name.

7 Directives

The two ‘\$’ characters appearing at the beginning of line (with optional preceding white-space) introduce special *preprocessor directives*.

This feature can be disabled using the `-Wno-directive` option (see [\[feature control\]](#), [page 11](#)).

The following directives are recognized:

```

$$source file [Directive]

```

```

$$include file [Directive]

```

Causes *file* to be read and processed at the current point. When the end of the file is reached, input is resumed from the previous input file.

If the file does not exist or cannot be read, `xenv` reports an error and exits with code 66 (see [Section 8.1 \[Exit codes\]](#), [page 13](#)).

Unless *file* is an absolute file name, it will be searched in the *include search path*. This path, initially empty, is initialized using the `-I` command line option. Its argument names a directory to be added to the include search path.

```

$$sinclude file [Directive]

```

Same as `$$include`, except that if the file is not found, no error is reported.

```

$$verbatim [Directive]

```

Begins a *verbatim text block*. The material up to the next `$$end` statement is reproduced on the output verbatim. Example:

```

    $$verbatim
    the $name construct is
    not expanded here
    $$end

```

See also the `$(...)` construct (see [\[inline verbatim\]](#), [page 3](#)), which introduces the same feature for inline portions of text.

```

$$ifdef name [Directive]

```

```

$$ifndef name [Directive]

```

Conditional expansion directive. The full syntax is:

```

$$ifdef name
text1
$$else
text2
$$endif

```

This construct is replaced with *text1*, if the environment variable *name* is defined, and with *text2* otherwise. The use of **\$\$ifndef** reverts the logic: *text1* is substituted if *name* is not defined.

```

$$ifset name [Directive]
$$ifnset name [Directive]

```

Another conditional expansion directive. The syntax is similar to the above:

```

$$ifset name
text1
$$else
text2
$$endif

```

text1 is substituted if *name* is defined and its value is not empty, otherwise *text2* is substituted.

The logic is inverted if **\$\$ifnset** is used instead.

Two more conditional directives are provided that analyze a *boolean* value of a variable. An environment variable is said to *evaluate to true* if its value is '1'. It is said to *evaluate to false* if its value is '0'. These default boolean constants can be changed using the **-Wbooleans** option (see [feature control], page 11).

The directives **\$\$iftrue** and **\$\$iffalse** control expansion using boolean evaluation:

```

$$iftrue name [Directive]

```

Conditional expansion based on boolean true. The syntax is similar to the above:

```

$$iftrue name
text1
$$else
text2
$$endif

```

If *name* is set and has a **true** boolean value, *text1* is substituted. Otherwise, if *name* is unset or evaluates to boolean **false**, *text2* is substituted.

```

$$iffalse name [Directive]

```

Conditional expansion based on boolean false. The syntax is:

```

$$iffalse name
text1
$$else
text2
$$endif

```

If *name* is unset or evaluates to boolean **false**, *text1* is substituted. Otherwise, if *name* evaluates to boolean **true**, *text2* is substituted.

`$$ifcom command` [Directive]
`$$ifncom command` [Directive]

Conditional expansion depending on the exit code of a shell command:

```

    $$ifcom command
    text1
    $$else
    text2
    $$endif

```

This directive runs *command* (via *SHELL -c*). If the exit code is 0, *text1* is substituted, otherwise *text2* is substituted.

`$$ifncom` inverts the logic (i.e. *text1* is substituted if *command* fails, i.e. exits with a non-zero status code).

A ‘`\newline`’ pair in *command* is treated as a line continuation: it is removed from the input stream and effectively ignored. This allows for splitting excessively long commands over multiple physical lines. To end command in a backslash, put two backslashes before the newline.

The *command* is run in the same manner as during command substitution (see [Chapter 4 \[Command substitution\], page 3](#)), except that its standard output is ignored. This means that it is passed to the shell verbatim (so that options `-u`, `-r`, and `-e` don’t apply) and its execution time is controlled by the `-t` option (see [\[command execution timeout\], page 11](#)).

In all conditional constructs described above the `$$else` part is optional. Conditional constructs can be nested to any depth.

`$$set name` [Directive]
 Sets the variable *name* to empty string.

`$$set name "value"` [Directive]
 Sets the variable *name* to *value*. *value* can occupy multiple lines and is subject to variable expansion, command substitution, and directive expansion, insofar as permitted by the `-W` option.

`$$set name 'value'` [Directive]
 Sets the variable *name* to *value*. *value* can occupy multiple lines. Neither variable expansion nor command substitution occurs in it.

`$$unset name` [Directive]
 Unsets the variable *name*.

7.1 Diagnostic messages and forced exit

`$$error` [Directive]
 This directive causes `xenv` to report a fatal error. Rest of the line following the whitespace after ‘`$$error`’ is used as the error message. Notice, that there’s no need to quote the message. E.g.:

```
$$error Something wrong happened
```

Processing resumes after the `$$error` directive. At the end of input, `xenv` exits with the code 65, unless exit code is altered by another error (see [Section 8.1 \[Exit codes\]](#), page 13).

```
$$warning [Directive]
```

Report a warning. Rest of the line following ‘`$$warning`’ and whitespace after it is used as the message text. Exit code is not altered. Processing resumes at the next line.

```
$$exit n [Directive]
```

Exit immediately with status *n*. If *n* is omitted, exit code is determined by usual rules. For example, if you want to emit an error message and stop further processing immediately, use:

```
$$error Unrecoverable error
$$exit
```

The exit status will be 65. Otherwise, if you want to exit with status 1:

```
$$error Unrecoverable error
$$exit 1
```

7.2 Looping directives

`xenv` provides two looping constructs: `$$loop` and `$$range`. Both allow you to repeat a block of text multiple times on the output.

The `$$loop` directive implements a *foreach loop*. The syntax is:

```
$$loop name args...
text
$$end
```

Here, *args* is a whitespace-delimited list of words. When the construct is scanned, *args* is subject to all kinds of expansions and *text* is scanned verbatim. During expansion, the `$$loop` directive will assign each value from this list to the variable *name* and expand *text*. Whatever value *name* has before entering the loop will be restored after leaving it.

For example, the following construct:

```
$$loop X A B C
This is $X
$$end
```

will produce the following expansion:

```
This is A
This is B
This is C
```

The `$$range` statement implements a *for loop*:

```
$$range name start stop [incr]
text
$$end
```

The three arguments are subject to expansions, whereas *text* is scanned verbatim. During expansion, the variable *name* is initialized with the value *start* and *text* is expanded. Then, the value of *name* is incremented by the value of *incr* and expansions continue until the *stop* value is reached (inclusive).

If *incr* is not given explicitly, +1 is assumed if *stop* > *start* and -1 is assumed otherwise.

For example:

```
$$range X 1 4
Number $X
$$end
```

expands to:

```
Number 1
Number 2
Number 3
Number 4
```

7.3 Eval

Sometimes you will need to compose the name of a variable and get its value. Such a need may arise, for example, when writing some complex loop. This is possible using the `$$eval` block.

`$$eval` [Directive]

Marks start of the text that should be evaluated after expansion. The end is marked with `$$end`. The text between `$$eval` and `$$end` is read and expanded, and then processed again. It is this second pass that creates output.

To illustrate that, suppose that the following variables are defined:

```
X2=two
I=2
```

and the following input is given to `xenv`:

```
$$eval
The value is \$$X$I.
$$end
```

This directive is processed in two passes. On the first pass, the text between `$$eval` and `$$end` is expanded. This pass produces:

```
The value is $X2
```

This text is stored away and expanded second time, which produces the following expansion:

```
The value is two.
```

When writing evaluation blocks, care should be taken to escape everything that should be expanded in the second pass. If the block is large, this can be a tedious process.

The following loop will consider variables `VAR_0` through `VAR_7` and print those of them that are defined:

```

$$range I 0 7
$$eval
\$\$ifset VAR_$I
Expand VAR_$I = \${VAR_$I};
\$\$endif
$$end
$$end

```

7.4 Diversions

Diversions are a way of directing output to a temporary storage and inserting it (*undiverting*) into the main output stream again at a later time. Temporary storage used for each diversion is identified by a unique *identifier* assigned to it when a diversion is declared.

\$\$divert [*name*] [Directive]

Begin diverting output to the diversion *name*. The diversion remains in effect until a subsequent **\$\$divert** directive or end of file is encountered, whichever happens first.

If *name* is omitted, main output stream is restored.

Several calls of **divert** with the same argument do not overwrite the previous diverted text, but append to it.

The diverted text remains in the temporary storage until it is discarded by **\$\$dropdivert**.

\$\$undivert *name* [Directive]

Undiverts the named diversion. Undiverted text is read verbatim. Several calls to **\$\$undivert** with the same argument are allowed. This provides a way to insert same text several times.

\$\$dropdivert *name* [Directive]

Discard and drop the diversion *name*.

M4 users should notice the following differences between diversions in **xenv** and **m4**:

1. Diversions are not implicitly undiverted at the end of input. Rather, they must be undiverted explicitly, when needed. Any diverted text that was not explicitly undiverted will be discarded.
2. Diverted text continues to exist after call to **\$\$undivert**. **\$\$undivert** can be used multiple times (even in a loop, if the need be).
3. To clear a diversion **\$\$dropdivert** should be used.

An example of using diversions:

```

begin text
$$divert A
this text is diverted first
$$divert B
this text is diverted after it
$$divert
main output restored
$$undivert B
some other text
$$undivert A

```

This will produce:

```

begin text
main output restored
this text is diverted after it
some other text
this text is diverted first

```

8 Invocation

The general usage is:

```
xenv [options] [file...]
```

The input is read from each *file* in turn, processed and the expansion is printed on the standard output. If no *file* arguments are given, the input is read from the standard input. Special file name ‘-’ (dash), stands for standard input as well.

Command line *options* are:

- e *name* Define the environment meta-variable *name*. See [Chapter 6 \[Meta-variable\]](#), [page 3](#).
- h
- ? Print a short command line summary and exit.
- m Pipe output to m4. See [GNU M4 macro processor](#). If the -s option is also given, it will be passed to m4 as well. See also -p.
- n *Dry-run* mode. Process input files without producing any output. Report any errors. This option is useful together with the -u to discover undefined variables.
- p *command* Pipe output to *command*.
- r Retain references to undefined variables in output. When this option is given any reference in the form ‘*\$name*’ or ‘*\${name}*’ that appears in the input text and refers to undefined variable *name* will be reproduced in output verbatim. By default, such references produce no expansion.
 Notice the emphasized text above. This option does not apply to variable references appearing in command substitutions (see [Chapter 4 \[Command substitution\]](#), [page 3](#)) and external tests (see [\[external tests\]](#), [page 6](#)).
 See also -u below.

- s** Generate synchronization directives, i.e. lines of the form
- `#line num "file"`
- which mean that the line that follows originated at line *num* in file *file*. Synchronization directives are emitted when variable or preprocessor directive expansion causes removal or insertion of one or more lines to the output. Each synchronization directive occupies a single line by itself. If a synchronization discrepancy occurs in the middle of an output line, emission of the synchronization directive is delayed until the next newline that does not occur in the middle of a quoted string (both single and double quotes are honored).
- t n** Sets maximum execution time for external commands to *n* seconds. The default is unlimited. This affects command substitution (see [Chapter 4 \[Command substitution\], page 3](#)) and `$$ifcom ($$ifncom)` conditionals (see [\[external tests\], page 6](#)).
- u** Treat unset variables as an error. Any `'$name'` or `'${name}'` construct *appearing in the input text* will trigger an error message, if the variable *name* is not defined. The offending construct will either produce no expansion or will be retained in the output, if the `-r` option is given. The program will continue running. When the input text is exhausted, it will exit with code 65 (see [Section 8.1 \[Exit codes\], page 13](#)).
- Notice the emphasized text above. This option does not apply to variable references appearing in command substitutions (see [Chapter 4 \[Command substitution\], page 3](#)) and external tests (see [\[external tests\], page 6](#)).
- D name[=value]**
- Define environment variable *name* to *value*, or to an empty string, if *value* is omitted.
- U name** Undefine the environment variable *name*.
- v** Print program version and exit.
- W [no-]feature**
- W feature=value**
- Controls the specific `xenv` *feature*. The first form enables or disable (if prefixed with `'no-'`) the feature. The second form sets the feature value. Valid *feature* names are:
- | | |
|---|-----------|
| command | [feature] |
| Controls command substitution (see Chapter 4 [Command substitution], page 3). | |
| comment | [feature] |
| Controls comments (see Chapter 5 [Comments], page 3). | |
| booleans=value | [feature] |
| This feature defines strings which are considered boolean values in <code>\$\$iftrue</code> and <code>\$\$iffalse</code> directives (see [boolean evaluation], page 5). The argument <i>value</i> is a comma-delimited list of <code>'t/f'</code> pairs. Each such | |

pair defines *t* as *true value* and *f* as *false value*. Either part may be omitted. For example:

```
-Wbooleans=1/0,true/false
```

This instructs `xenv` to evaluate ‘1’ and ‘true’ as boolean `true` and ‘0’ or ‘false’ as boolean `false`.

directive [feature]

Preprocessor directives (see [Chapter 7 \[Directives\]](#), page 4).

escape [feature]

Use of ‘\’ as escape character (see [Chapter 3 \[Escapes\]](#), page 2).

minimal [feature]

Disables most other features and enforces `$ENV` as environment meta-variable (see [Chapter 6 \[Meta-variable\]](#), page 3). In other words, `-Wminimal` is equivalent to:

```
-Wno-command -Wno-comment -Wno-directive -Wno-quote \  
-Wno-escape -e ENV
```

paranoid-memfree [feature]

Free all allocated memory before exit. It is needed mostly for debugging and chasing memory allocation errors.

stashfiles=*n* [feature]

Sets allowed number of simultaneously open stash files. A *stash* is an internal structure used to store temporary data. Stashes are used, in particular, to implement diversions, loops and conditional constructs. A stash attempts to store as much data as possible in the RAM. The amount of memory allotted for each stash is defined by the `stashsize` feature (see below). When its memory gets filled, it opens a temporary file and stores the surplus data there. When a new stash file needs to be open and the number of already open files equals *n*, the program will close the least recently used stash file, thereby freeing a file descriptor for use of the new stash. The closed file will eventually be reopened when it is needed.

By default, the number of stashfiles is set to 2/3 of the system limit on the number of open file descriptors. You may wish to set this feature if `xenv` terminates with a ‘too many open files’ error.

stashsize=*n* [feature]

Sets amount of memory allocated per stash. When this memory is filled, a temporary file is created and all surplus data are stored there. The *n* can be followed by the usual suffixes: ‘K’, ‘M’, and ‘G’. Lower-case letters are allowed too.

The default memory size is set to the system memory page size.

quote [feature]

Controls inline verbatim quotations (see [\[inline verbatim\]](#), page 3).

By default all features except `minimal` are enabled.

-x Enable debugging output.

8.1 Exit codes

0	Successful termination.
64	Command line usage error.
65	Reference to undefined variable encountered when run with the <code>-u</code> option.
66	Source file does not exist.
69	Recursive inclusion has been prevented.
70	Internal program error. This probably indicates a bug. Please, report it. see Chapter 10 [Bug reports] , page 13.
71	System error: file cannot be opened, fork failed, etc.
77	Permission denied when trying to open the source file.

9 Downloads and Installation

The program can be downloaded from <https://download.gnu.org.ua/release/xenv>.

Docker images with `xenv` and GNU pies (see *GNU pies*) are available from <https://hub.docker.com/repository/docker/graygnuorg/pies>.

When installing from sources, the procedure is as simple as:

```
./configure
make
make install
```

For a detailed discussion of the installation procedure, [Section “Running configure Scripts”](#) in *Autoconf*.

10 Bug Reports

If you think you found a bug in `xenv` or in its documentation, please send a mail to gray@gnu.org (Sergey Poznyakoff) or use the bug tracker at <https://puszcza.gnu.org.ua/bugs/?group=xenv> (requires authorization).

Appendix A GNU Free Documentation License

Version 1.3, 3 November 2008

Copyright © 2000, 2001, 2002, 2007, 2008 Free Software Foundation, Inc.
<https://fsf.org/>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document *free* in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or non-commercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of “copyleft”, which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The “Document”, below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as “you”. You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A “Modified Version” of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A “Secondary Section” is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document’s overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The “Invariant Sections” are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The “Cover Texts” are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A “Transparent” copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not “Transparent” is called “Opaque”.

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The “Title Page” means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, “Title Page” means the text near the most prominent appearance of the work’s title, preceding the beginning of the body of the text.

The “publisher” means any person or entity that distributes copies of the Document to the public.

A section “Entitled XYZ” means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as “Acknowledgements”, “Dedications”, “Endorsements”, or “History”.) To “Preserve the Title” of such a section when you modify the Document means that it remains a section “Entitled XYZ” according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.

- D. Preserve all the copyright notices of the Document.
- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- H. Include an unaltered copy of this License.
- I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.
- N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.
- O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be

added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled “History” in the various original documents, forming one section Entitled “History”; likewise combine any sections Entitled “Acknowledgements”, and any sections Entitled “Dedications”. You must delete all sections Entitled “Endorsements.”

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an “aggregate” if the copyright resulting from the compilation is not used to limit the legal rights of the compilation’s users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document’s Cover Texts may be placed on covers that bracket the Document within the aggregate, or the

electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled “Acknowledgements”, “Dedications”, or “History”, the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <https://www.gnu.org/licenses/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License “or any later version” applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free

Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

11. RELICENSING

“Massive Multiauthor Collaboration Site” (or “MMC Site”) means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A “Massive Multiauthor Collaboration” (or “MMC”) contained in the site means any set of copyrightable works thus published on the MMC site.

“CC-BY-SA” means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

“Incorporate” means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is “eligible for relicensing” if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

```
Copyright (C) year your name.
Permission is granted to copy, distribute and/or modify this document
under the terms of the GNU Free Documentation License, Version 1.3
or any later version published by the Free Software Foundation;
with no Invariant Sections, no Front-Cover Texts, and no Back-Cover
Texts. A copy of the license is included in the section entitled ‘‘GNU
Free Documentation License’’.
```

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the “with...Texts.” line with this:

```
with the Invariant Sections being list their titles, with
the Front-Cover Texts being list, and with the Back-Cover Texts
being list.
```

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

Index

\$

<code>\$\$divert</code>	9
<code>\$\$dropdivert</code>	9
<code>\$\$else</code>	4
<code>\$\$end</code>	4
<code>\$\$endif</code>	4
<code>\$\$error</code>	6
<code>\$\$eval</code>	8
<code>\$\$exit</code>	7
<code>\$\$ifcom</code>	6
<code>\$\$ifdef</code>	4
<code>\$\$iffalse</code>	5
<code>\$\$ifncom</code>	6
<code>\$\$ifndef</code>	4
<code>\$\$ifnset</code>	5
<code>\$\$ifset</code>	5
<code>\$\$iftrue</code>	5
<code>\$\$include</code>	4
<code>\$\$loop</code>	7
<code>\$\$range</code>	7
<code>\$\$set</code>	6
<code>\$\$sinclude</code>	4
<code>\$\$source</code>	4
<code>\$\$undivert</code>	9
<code>\$\$unset</code>	6
<code>\$\$verbatim</code>	4
<code>\$\$warning</code>	7

-

<code>-e</code>	2, 3
<code>-I</code>	4
<code>-r</code>	2
<code>-u</code>	2
<code>-Wno-command</code>	3
<code>-Wno-comment</code>	3
<code>-Wno-directive</code>	4
<code>-Wno-escape</code>	2
<code>-Wno-quote</code>	3

B

backslash, disable escape character	2
boolean evaluation	5
<code>booleans=value</code>	11

C

<code>command</code>	11
command substitutions, disable	3
<code>comment</code>	11
comments, disable	3
conditional expansion	4, 5

D

<code>directive</code>	12
directives	4
directives, disabling	4

disable command substitutions	3
disable comments	3
disable directives	4
disable escape sequence	2
disabling inline verbatim text	3
dry run	10

E

<code>EMFILE</code>	12
<code>escape</code>	12
escaping, disable	2

F

feature control	11
-----------------------	----

I

include search path	4
inline verbatim text	3
inline verbatim text, disable	3

M

<code>m4</code>	10
Meta-variable	3
<code>minimal</code>	12
<code>minimal</code> , feature	12

O

open files	12
------------------	----

P

<code>paranoid-memfree</code>	12
preprocessor directives	4

Q

<code>quote</code>	12
--------------------------	----

S

<code>stash</code>	12
<code>stashfiles=n</code>	12
<code>stashsize=n</code>	12
synchronization directive	11

T

Too many open files	12
---------------------------	----

V

verbatim text block	4
---------------------------	---